

УДК 621.326

Якубів П.–ст. гр. СНм-51

Тернопільський національний технічний університет імені Івана Пулюя

EMBER.JS КОНЦЕПЦІЇ ДЛЯ RUBY ON RAILS РОЗРОБНИКА

Науковий керівник: к.т.н., доцент Загородна Н.В.

Yakubiv P.

Ternopil Ivan Pul'uj National Technical University

EMBER.JS CONCEPTS FOR RUBY ON RAILS DEVELOPER

Supervisor: Zahorodna N.

Ключові слова: Ruby, Ruby on Rails, Ember

Keywords: Ruby, Ruby on Rails, Ember

Перехід від багатофункціонального (full stack) фреймворка, такого як Ruby on Rails (RoR), до клієнтського, як Ember.js, є не таким складним, як видається на перший погляд. Ember.js надає програмісту нові інструменти розробки, більшість з яких є аналогічними з RoR інструментами. Але виникає питання, як поєднати ці інструменти в єдине ціле, забезпечивши повноцінне функціонування веб-додатку. Як і Rails, Ember базується на Model-View-Controller (MVC) концепції проектування.

Компоненти Model в Rails та Ember суттєво не відрізняються і служать для визначення структур даних та встановлення асоціацій між іншими структурами. Під структурою розуміють будь-яку сутність з набором атрибутів, яка в подальшому буде зберігатися в базі даних.

Компонента View в Rails і Ember використовується для генерування інтерфейсу додатку. У RoR View - це HTML Template, який представляє контент з бази даних за допомогою ERB або HAML, використовуючи при цьому допоміжні класи для забезпечення основного Rails патерну DRY (Don't Repeat Yourself). В Ember є суттєва різниця між View і Template. Template генеруються за допомогою Handlebars, що є синонімом до ERB і HAML в Rails. View представляє логіку, яка керує обробкою подій, безпосередньо перед представленням елементів інтерфейсу.

Кожному RoR розробнику відомо, що Controller здійснює: відправку інформації в View з бази даних, перенаправляє запити в браузері, створює або видаляє об'єкти бази даних. Під час роботи з Controllers в клієнтських фреймворках, Controller прийнято називати "View Controller", через те, що він контролює лише View. Controller в Ember відповідає за функціональні можливості призначені для користувацького інтерфейсу, які не зберігаються в базі даних. Наприклад, коли необхідно приховати той чи інший елемент інтерфейсу, достатньо лиш передати значення атрибута в Controller, після цього він передасть значення в Template, який оновить інтерфейс. В Ember.js Controller виконує роль додаткового проширення моделі, визначаючи як вона буде відображатися в Template. Controllers не завжди підтримуються моделями, в деяких випадках, вони створюються для зберігання атрибутів, які мають відношення до поточного View.

Отже, кожному Ruby on Rails розробнику, стає все більше необхідно мати практичні навички роботи з будь-яким JavaScript клієнтським фреймворком (Ember.js, Angular, Backbone), щонайменше, для забезпечення швидкої взаємодії користувача з веб-додатком.